

Group members: Eduard Koshkelyan, Roshan Maharjan, Marwane Bidamou

Lab W1D5

Question 1. Aim of this question is to understand amortized cost analysis.

(a) Show all the calculations:

Sample Instance 4 of Clearable Table

add, add, add, add, clear, add, clear, add, clear, add, add, add, clear, add, add, clear, add, add, clear.

Answer:

Each add operation costs 1, and each clear operation costs the number of items present in the table in the specific moment

So, here's the calculation step by step:

Normal calculations :

add (+1 => 1), add (+1 => 2), add (+1 => 3), add (+1 => 4), clear (+4 => 8), add (+1 => 9), clear (+1 => 10), add (+1 => 11), clear (+1 => 12), add (+1 => 13), add (+1 => 14), add (+1 => 15), clear (+3 => 18), add (+1 => 19), add (+1 => 20), clear (+2 => 22), add (+1 => 23), add (+1 => 24), clear (+2 => 26), add (+1 => 27), add (+1 => 28), clear (+2 => 30).

Amortized calculations:

add (+2 => 2), add (+2 => 4), add (+2 => 6), add (+2 => 8), clear (+0 => 8), add (+2 => 10), clear (+0 => 10), add (+2 => 12), clear (+0 => 12), add (+2 => 14), add (+2 => 16), add (+2 => 18), clear (+0 => 18), add (+2 => 20), add (+2 => 22), clear (+0 => 22), add (+2 => 24), add (+2 => 26), clear (+0 => 26), add (+2 => 28), add (+2 => 30), clear (+0 => 30).

Total cost: 30

Total amortized cost: 30

Number of operations: 22

Average cost per operation: $30/22 = 1.36 \leq 2$

(b) Show all the calculations:

Sample Instance 3 of ArrayList with size doubling strategy

A resize just happened from size 16 to size 32.

Answer:

So after the resize from 16 to 32, we now have 16 free spaces in the new ArrayList

And each add operation costs 1, and resize operation costs $3k$, where k is the new size of array

Resize cost = $32 * 3 = 96$

Total cost = $96 + 16 = 112$

Amortized Cost (add) = $112/16 = 7$

Amortized Cost(resize) = 0

Question 2. Aim of this question is to better understand amortized cost analysis.

Data structure : ArrayList with size tripling strategy.

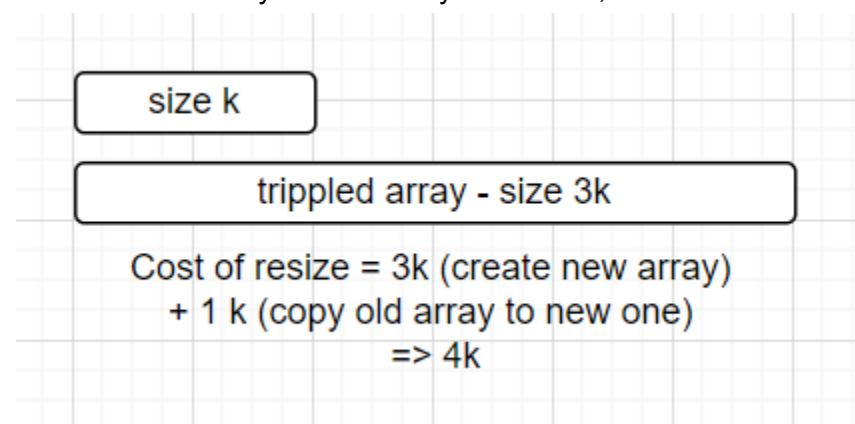
Answer all questions below giving detailed explanation.

(a) What is the actual cost of add?

Answer: The actual cost of add is 1 if there is available space in the array, otherwise if the array is full and a resize is required before adding then the add operation includes the cost of resize plus the cost of add.

(b) What is the actual cost of resize?

Answer: The actual cost of resize is $4k$, where k is the number of items in the ArrayList, in details it costs $3k$ to create a new array of size $3*k$, and then it costs k to copy the current array content to the newly created array. So in total, it's $3k + k = 4k$



(c) Using traditional worst-case analysis, show that the average cost of an operation is **NOT constant time.**

Answer:

We can take add operation as an example

- The cost of an add operation without resizing is 1, as it simply places it in the new available slot in the array. This is a constant-time operation, $O(1)$
- The cost of an add operation without resizing: since the array is filled, a resize operation is triggered. In this case the add operation has to include the cost of resizing the array plus the cost of adding the element, so it will be $3k+1$

(d) Consider a sample instance (hint : resize just happened and current size of the array is 9. (You should never consider current size = 1 for this type of calculation). You are adding. Then you resized again)

A. What is the Amortized_Cost(add)?

- So after the resize from 3 to 9, we now have 6 free spaces in the new array, and each operation costs 1, and resize operation costs $4k$
- Resize cost = $4k = 4*9 = 36$
- Total cost = $36 + 6$ (free space) = 42

- Amortized cost(add) = Total amortized cost / The number of operations
= 42 / 6 = 7

B. What is the Amortized_Cost(resize)?

- Once we have resized the array, we have accounted for the resize cost, and it has been distributed across the **add** operations. Therefore, the amortized cost of resize is effectively 0 after it has been distributed.

C. Through amortized cost analysis show if there is sequence of n operations (some add, some resize) the average cost of an operation is constant time.

- total amortized cost = 7n.
- The number of operations = n.
- The average cost = Total amortized cost / The number of operations = 7n / n = 7.
The average cost of an operation is 7. The average cost of an operation is constant time!